WEST

Freeform Search

Database:
US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term:

Display: 10 **Documents in Display Format:** KWIC **Starting with Number** 1

Generate: ○ Hit List ◉ Hit Count ○ Side by Side ○ Image

Search | Clear | Help | Logout | Interrupt

Main Menu | Show S Numbers | Edit S Numbers | Preferences | Cases

Search History

DATE: Friday, October 17, 2003 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L5	L3 and (torus)	1	L5
L4	L3 and (torus adj1 direction$)	0	L4
L3	L2 and (cluster adj1 switch$)	1	L3
L2	L1 and cluster$	1	L2
L1	6338129.pn.	1	L1

END OF SEARCH HISTORY

WEST

Generate Collection | Print

Search Results - Record(s) 1 through 1 of 1 returned.

☐ 1. Document ID: US 6338129 B1

L5: Entry 1 of 1 File: USPT Jan 8, 2002

DOCUMENT-IDENTIFIER: US 6338129 B1
TITLE: Manifold array processor

Abstract Text (1):
An array processor includes processing elements arranged in clusters which are, in turn, combined in a rectangular array. Each cluster is formed of processing elements which preferably communicate with the processing elements of at least two other clusters. Additionally each inter-cluster communication path is mutually exclusive, that is, each path carries either north and west, south and east, north and east, or south and west communications. Due to the mutual exclusivity of the data paths, communications between the processing elements of each cluster may be combined in a single inter-cluster path. That is, communications from a cluster which communicates to the north and east with another cluster may be combined in one path, thus eliminating half the wiring required for the path. Additionally, the length of the longest communication path is not directly determined by the overall dimension of the array, as it is in conventional torus arrays. Rather, the longest communications path is limited only by the inter-cluster spacing. In one implementation, transpose elements of an N.times.N torus are combined in clusters and communicate with one another through intra-cluster communications paths. Since transpose elements have direct connections to one another, transpose operation latency is eliminated in this approach. Additionally, each PE may have a single transmit port and a single receive port. As a result, the individual PEs are decoupled from the topology of the array.

US Patent No. (1):
6338129

Brief Summary Text (7):
In the nearest neighbor torus connected computer of FIG. 1A multiple processing elements (PEs) are connected to their north, south, east and west neighbor PEs through torus connection paths MP and all PEs are operated in a synchronous single instruction multiple data (SIMD) fashion. Since a torus connected computer may be obtained by adding wraparound connections to a mesh-connected computer, a mesh-connected computer, one without wraparound connections, may be thought of as a subset of torus connected computers. As illustrated in FIG. 1B, each path MP may include T transmit wires and R receive wires, or as illustrated in FIG. 1C, each path MP may include B bidirectional wires. Although unidirectional and bidirectional communications are both contemplated by the invention, the total number of bus wires, excluding control signals, in a path will generally be referred to as k wires hereinafter, where k=B in a bidirectional bus design and k=T+R in a unidirectional bus design. It is assumed that a PE can transmit data to any of its neighboring PEs, but only one at a time. For example, each PE can transmit data to its east neighbor in one communication cycle. It is also assumed that a broadcast mechanism is present such that data and instructions can be dispatched from a controller simultaneously to all PEs in one broadcast dispatch period.

Brief Summary Text (8):
Although bit-serial inter-PE communications are typically employed to minimize wiring complexity, the wiring complexity of a torus-connected array nevertheless presents implementation problems. The conventional torus-connected array of FIG. 1A

includes sixteen processing elements connected in a four by four array 10 of PEs. Each processing element PE.sub.i,j is labeled with its row and column number i and j, respectively. Each PE communicates to its nearest North (N), South (S), East (E) and West (W) neighbor with point to point connections. For example, the connection between PE.sub.0,0 and PE.sub.3,0 shown in FIG. 1A is a wraparound connection between PE.sub.0,0 's N interface and PE.sub.3,0 's south interface, representing one of the wraparound interfaces that forms the array into a torus configuration. In such a configuration, each row contains a set of N interconnections and, with N rows, there are N.sup.2 horizontal connections. Similarly, with N columns having N vertical interconnections each, there are N.sup.2 vertical interconnections. For the example of FIG. 1A, N=4. The total number of wires, such as the metallization lines in an integrated circuit implementation in an N.times.N torus-connected computer including wraparound connections, is therefore 2kN.sup.2, where k is the number of wires in each interconnection. The number k may be equal to one in a bit serial interconnection. For example with k=1 for the 4.times.4 array 10 as shown in FIG. 1A, 2kN.sup.2 =32.

Brief Summary Text (9):
For a number of applications where N is relatively small, it is preferable that the entire PE array is incorporated in a single integrated circuit. The invention does not preclude implementations where each PE can be a separate microprocessor chip, for example. Since the total number of wires in a torus connected computer can be significant, the interconnections may consume a great deal of valuable integrated circuit "real estate", or the area of the chip taken up. Additionally, the PE interconnection paths quite frequently cross over one another complicating the IC layout process and possibly introducing noise to the communications lines through crosstalk. Furthermore, the length of wraparound links, which connect PEs at the North and South and at the East and West extremes of the array, increase with increasing array size. This increased length increases each communication line's capacitance, thereby reducing the line's maximum bit rate and introducing additional noise to the line.

Brief Summary Text (10):
Another disadvantage of the torus array arises in the context of transpose operations. Since a processing element and its transpose are separated by one or more intervening processing elements in the communications path, latency is introduced in operations which employ transposes. For example, should the PE.sub.2,1 require data from its transpose, PE.sub.1,2, the data must travel through the intervening PE.sub.1,1 or PE.sub.2,2. Naturally, this introduces a delay into the operation, even if PE.sub.1,1 and PE.sub.2,2 are not otherwise occupied. However, in the general case where the PEs are implemented as micro-processor elements, there is a very good probability that PE.sub.1,1 and PE.sub.2,2 will be performing other operations and, in order to transfer data or commands from PE.sub.1,2 to PE.sub.2,1, they will have to set aside these operations in an orderly fashion. Therefore, it may take several operations to even begin transferring the data or commands from PE.sub.1,2 to PE.sub.1,1 and the operations PE.sub.1,1 was forced to set aside to transfer the transpose data will also be delayed. Such delays snowball with every intervening PE and significant latency is introduced for the most distant of the transpose pairs. For example the PE.sub.3,1 /PE.sub.1,3 transpose pair of FIG. 1A, has a minimum of three intervening PEs, requiring a latency of four communication steps and could additionally incur the latency of all the tasks which must be set aside in all those PEs in order to transfer data between PE.sub.3,1 and PE.sub.1,3 in the general case.

Brief Summary Text (11):
Recognizing such limitations of torus connected arrays, new approaches to arrays have been disclosed in U.S. Pat. No. 5,612,908; A Massively Parallel Diagonal Fold Array Processor, G. G. Pechanek et al., 1993 International Conference on Application Specific Array Processors, pp. 140-143, Oct. 25-27, 1993, Venice, Italy, and Multiple Fold Clustered Processor Torus Array, G. G. Pechanek, et. al., Proceedings Fifth NASA Symposium on VLSI Design, pp. 8.4.1-11, Nov. 4-5, 1993, University of New Mexico, Albuquerque, N. Mex. which are incorporated by reference herein in their entirety. The operative technique of these torus array organizations is the folding of arrays of PEs using the diagonal PEs of the conventional nearest neighbor torus as the foldover edge. As illustrated in the array 20 of FIG. 2, these techniques may

be employed to substantially reduce inter-PE wiring, to reduce the number and length of wraparound connections, and to position PEs in close proximity to their transpose PEs. This processor array architecture is disclosed, by way of example, in U.S. Pat. Nos. 5,577,262, 5,612,908, and EP 0,726,532 and EP 0,726,529 which were invented by the same inventor as the present invention and are incorporated herein by reference in their entirety. While such arrays provide substantial benefits over the conventional torus architecture, due to the irregularity of PE combinations, for example in a single fold diagonal fold mesh, some PEs are clustered "in twos", others are single, in a three fold diagonal fold mesh there are clusters of four PEs and eight PEs. Due to an overall triangular shape of the arrays, the diagonal fold type of array presents substantial obstacles to efficient, inexpensive integrated circuit implementation. Additionally, in a diagonal fold mesh as in EP 0,726,532 and EP 0,726,529, and other conventional mesh architectures, the interconnection topology is inherently part of the PE definition. This fixes the PE's position in the topology, consequently limiting the topology of the PEs and their connectivity to the fixed configuration that is implemented. Thus, a need exists for further improvements in processor array architecture and processor interconnection.

Brief Summary Text (13):
The present invention is directed to an array of processing elements which substantially reduce the array's interconnection wiring requirements when compared to the wiring requirements of conventional torus processing element arrays. In a preferred embodiment, one array in accordance with the present invention achieves a substantial reduction in the latency of transpose operations. Additionally, the inventive array decouples the length of wraparound wiring from the array's overall dimensions, thereby reducing the length of the longest interconnection wires. Also, for array communication patterns that cause no conflict between the communicating PEs, only one transmit port and one receive port are required per PE, independent of the number of neighborhood connections a particular topology may require of its PE nodes. A preferred integrated circuit implementation of the array includes a combination of similar processing element clusters combined to present a rectangular or square outline. The similarity of processing elements, the similarity of processing element clusters, and the regularity of the array's overall outline make the array particularly suitable for cost-effective integrated circuit manufacturing.

Brief Summary Text (14):
To form an array in accordance with the present invention, processing elements may first be combined into clusters which capitalize on the communications requirements of single instruction multiple data ("SIMD") operations. Processing elements may then be grouped so that the elements of one cluster communicate within a cluster and with members of only two other clusters. Furthermore, each cluster's constituent processing elements communicate in only two mutually exclusive directions with the processing elements of each of the other clusters. By definition, in a SIMD torus with unidirectional communication capability, the North/South directions are mutually exclusive with the East/West directions. Processing element clusters are, as the name implies, groups of processors formed preferably in close physical proximity to one another. In an integrated circuit implementation, for example, the processing elements of a cluster preferably would be laid out as close to one another as possible, and preferably closer to one another than to any other processing element in the array. For example, an array corresponding to a conventional four by four torus array of processing elements may include four clusters of four elements each, with each cluster communicating only to the North and East with one other cluster and to the South and West with another cluster, or to the South and East with one other cluster and to the North and West with another cluster. By clustering PEs in this manner, communications paths between PE clusters may be shared, through multiplexing, thus substantially reducing the interconnection wiring required for the array.

Brief Summary Text (15):
In a preferred embodiment, the PEs comprising a cluster are chosen so that processing elements and their transposes are located in the same cluster and communicate with one another through intra-cluster communications paths, thereby eliminating the latency associated with transpose operations carried out on conventional torus arrays. Additionally, since the conventional wraparound path is

treated the same as any PE-to-PE path, the longest communications path may be as short as the inter-cluster spacing, regardless of the array's overall dimension. According to the invention an N.times.M torus may be transformed into an array of M clusters of N PEs, or into N clusters of M PEs.

Drawing Description Text (2):
FIG. 1A is a block diagram of a conventional prior art 4.times.4 nearest neighbor connected torus processing element (PE) array;

Drawing Description Text (3):
FIG. 1B illustrates how the prior art torus connection paths of FIG. 1A may include T transmit and R receive wires;

Drawing Description Text (4):
FIG. 1C illustrates how prior art torus connection paths of FIG. 1A may include B bidirectional wires;

Drawing Description Text (8):
FIG. 4 is a tiling of a 4.times.4 torus which illustrates all the torus's inter-PE communications links;

Drawing Description Text (9):
FIGS. 5A through 5G are tilings of a 4.times.4 torus which illustrate the selection of PEs for cluster groupings in accordance with the present invention;

Drawing Description Text (10):
FIG. 6 is a tiling of a 4.times.4 torus which illustrates alternative grouping of PEs for clusters;

Drawing Description Text (11):
FIG. 7 is a tiling of a 3.times.3 torus which illustrates the selection of PEs for PE clusters;

Drawing Description Text (12):
FIG. 8 is a tiling of a 3.times.5 torus which illustrates the selection of PEs for PE clusters;

Drawing Description Text (13):
FIG. 9 is a block diagram illustrating an alternative, rhombus/cylinder approach to selecting PEs for PE clusters;

Drawing Description Text (14):
FIG. 10 is a block diagram which illustrates the inter-cluster communications paths of the new PE clusters;

Drawing Description Text (15):
FIGS. 11A and 11B illustrate alternative rhombus/cylinder approaches to PE cluster selection;

Drawing Description Text (19):
FIGS. 15A through 15D are block diagram illustrations of inter-cluster communications paths for 3, 4, 5, and 6 cluster by 6 PE arrays, respectively;

Drawing Description Text (20):
FIG. 16 is a block diagram illustrating East/South communications paths within an array of four four-member clusters;

Drawing Description Text (21):
FIG. 17 is a block diagram illustration of East/South and West/North communications paths within an array of four four-member clusters;

Drawing Description Text (22):
FIG. 18 is a block diagram illustrating one of the clusters of the embodiment of FIG. 17, which illustrates in greater detail a cluster switch and its interface to the illustrated cluster;

Drawing Description Text (24):
FIGS. 19C and 19D are block diagrams which respectively illustrate a portion of an image within a 4.times.4 block and the block loaded into conventional torus locations; and

Detailed Description Text (2):
In one embodiment, a new array processor in accordance with the present invention combines PEs in clusters, or groups, such that the elements of one cluster communicate with members of only two other clusters and each cluster's constituent processing elements communicate in only two mutually exclusive directions with the processing elements of each of the other clusters. By clustering PEs in this manner, communications paths between PE clusters may be shared, thus substantially reducing the interconnection wiring required for the array. Additionally, each PE may have a single transmit port and a single receive port or, in the case of a bidirectional sequential or time sliced transmit/receive communication implementation, a single transmit/receive port. As a result, the individual PEs are decoupled from the topology of the array. That is, unlike a conventional torus connected array where each PE has four bidirectional communication ports, one for communication in each direction, PEs employed by the new array architecture need only have one port. In implementations which utilize a single transmit and a single receive port, all PEs in the array may simultaneously transmit and receive. In the conventional torus, this would require four transmit and four receive ports, a total of eight ports, per PE, while in the present invention, one transmit port and one receive port, a total of two ports, per PE are required.

Detailed Description Text (3):
In one presently preferred embodiment, the PEs comprising a cluster are chosen so that processing elements and their transposes are located in the same cluster and communicate with one another through intra-cluster communications paths. For convenience of description, processing elements are referred to as they would appear in a conventional torus array, for example, processing element PE.sub.0,0 is the processing element that would appear in the "Northwest" corner of a conventional torus array. Consequently, although the layout of the new cluster array is substantially different from that of a conventional array processor, the same data would be supplied to corresponding processing elements of the conventional torus and new cluster arrays. For example, the PE.sub.0,0 element of the new cluster array would receive the same data to operate on as the PE.sub.0,0 element of a conventional torus-connected array. Additionally, the directions referred to in this description will be in reference to the directions of a torus-connected array. For example, when communications between processing elements are said to take place from North to South, those directions refer to the direction of communication within a conventional torus-connected array.

Detailed Description Text (4):
The PEs may be single microprocessor chips that may be of a simple structure tailored for a specific application. Though not limited to the following description, a basic PE will be described to demonstrate the concepts involved. The basic structure of a PE 30 illustrating one suitable embodiment which may be utilized for each PE of the new PE array of the present invention is illustrated in FIG. 3A. For simplicity of illustration, interface logic and buffers are not shown. A broadcast instruction bus 31 is connected to receive dispatched instructions from a SIMD controller 29, and a data bus 32 is connected to receive data from memory 33 or another data source external to the PE 30. A register file storage medium 34 provides source operand data to execution units 36. An instruction decoder/controller 38 is connected to receive instructions through the broadcast instruction bus 31 and to provide control signals 21 to registers within the register file 34 which, in turn, provide their contents as operands via path 22 to the execution units 36. The execution units 36 receive control signals 23 from the instruction decoder/controller 38 and provide results via path 24 to the register file 34. The instruction decoder/controller 38 also provides cluster switch enable signals on an output the line 39 labeled Switch Enable. The function of cluster switches will be discussed in greater detail below in conjunction with the discussion of FIG. 18. Inter-PE communications of data or commands are received at receive input 37 labeled Receive and are transmitted from a transmit output 35

labeled Send.

Detailed Description Text (6):
A conventional 4.times.4 nearest neighbor torus of PEs of the same type as the PE 30 illustrated in FIG. 3A is shown surrounded by tilings of itself in FIG. 4. The center 4.times.4 torus 40 is encased by a ring 42 which includes the wraparound connections of the torus. The tiling of FIG. 4 is a descriptive aid used to "flatten out" the wraparound connections and to thereby aid in explanation of the preferred cluster forming process utilized in the array of one embodiment of the present invention. For example, the wraparound connection to the west from PE.sub.00, is PE.sub.0,3, that from the PE.sub.1,3 to the east is PE.sub.1,0, etc., as illustrated within the block 42. The utility of this view will be more apparent in relation to the discussion below of FIGS. 5A-5G.

Detailed Description Text (7):
In FIG. 5A, the basic 4.times.4 PE torus is once again surrounded by tilings of itself. The present invention recognizes that communications to the East and South from PE.sub.0,0 involve PE.sub.0,1 and PE.sub.1,0, respectively. Furthermore, the PE which communicates to the east to PE.sub.1,0 is PE.sub.1,3 and PE.sub.1,3 communicates to the South to PE.sub.2,3. Therefore, combining the four PEs, PE.sub.0,0, PE.sub.1,3, PE.sub.2,2, and PE.sub.3,1 in one cluster yields a cluster 44 from which PEs communicate only to the South and East with another cluster 46 which includes PEs, PE.sub.0,1, PE.sub.1,0, PE.sub.2,3 and PE.sub.3,2. Similarly, the PEs of cluster 46 communicate to the South and East with the PEs of cluster 48 which includes PEs, PE.sub.0,2, PE.sub.1,1, PE.sub.2,0, and PE.sub.3,3. The PEs, PE.sub.0,3, PE.sub.1,2, PE.sub.2,1, and PE.sub.3,0 of cluster 50 communicate to the South and East with cluster 44. This combination yields clusters of PEs which communicate with PEs in only two other clusters and which communicate in mutually exclusive directions to those clusters. That is, for example, the PEs of cluster 48 communicate only to the South and East with the PEs of cluster 50 and only to the North and West with the PEs of cluster 46. It is this exemplary of grouping of PEs which permits the inter-PE connections within an array in accordance with the present invention to be substantially reduced in comparison with the requirements of the conventional nearest neighbor torus array.

Detailed Description Text (8):
Many other combinations are possible. For example, starting again with PE.sub.0,0 and grouping PEs in relation to communications to the North and East yields clusters 52, 54, 56 and 58 of FIG. 5B. These clusters may be combined in a way which greatly reduces the interconnection requirements of the PE array and which reduces the length of the longest inter-PE connection. However, these clusters do not combine PEs and their transposes as the clusters 44-50 in FIG. 5A do. That is, although transpose pairs PE.sub.0,2 /PE.sub.2,0 and PE.sub.1,3 /PE.sub.3,1 are contained in cluster 56, the transpose pair PE.sub.0,1 /PE.sub.1,0 is split between clusters 54 and 58. An array in accordance with the presently preferred embodiment employs only clusters such as 44-50 which combine all PEs with their transposes within clusters. For example, in FIG. 5A the PE.sub.3,1 /PE.sub.1,3 transpose pair is contained within cluster 44, the PE.sub.3,2,PE.sub.2,3 and PE.sub.1,0 /PE.sub.0,1 transpose pairs are contained within cluster 46, the PE.sub.0,2 /PE.sub.2,0 transpose pair is contained within cluster 48, and the PE.sub.3,0 /PE.sub.0,3 and PE.sub.2,1 /PE.sub.1,2 transpose pairs are contained within cluster 50. Clusters 60, 62, 64 and 68 of FIG. 5C are formed, starting at PE.sub.0,0, by combining PEs which communicate to the North and West. Note that cluster 60 is equivalent to cluster 44, cluster 62 is equivalent to cluster 46, cluster 64 is equivalent to cluster 48 and cluster 68 is equivalent to cluster 50. Similarly, clusters 70 through 76 of FIG. 5D, formed by combining PEs which communicate to the South and West, are equivalent to clusters 52 through 58, respectively of FIG. 5B. As demonstrated in FIG. 5E, clusters 45, 47, 49 and 51, which are equivalent to the preferred clusters 48, 50, 44 and 46 may be obtained from any "starting point" within the torus 40 by combining PEs which communicate to the South and East.

Detailed Description Text (9):
Another clustering is depicted in FIG. 5F where clusters 61, 63, 65, and 67 form a criss cross pattern in the tilings of the torus 40. This clustering demonstrates that there are a number of ways in which to group PEs to yield clusters which

communicate with two other clusters in mutually exclusive directions. That is, PE.sub.0,0 and PE.sub.2,2 of cluster 65 communicate to the East with PE.sub.0,1 and PE.sub.2,3, respectively, of cluster 61. Additionally, PE.sub.1,1 and PE.sub.3,3 of cluster 65 communicate to the West with PE.sub.1,0 and PE.sub.3,2, respectively, of cluster 61. As will be described in greater detail below, the Easterly communications paths just described, that is, those between PE.sub.0,0 and PE.sub.0,1 and between PE.sub.2,2 and PE.sub.2,3 and other inter-cluster paths may be combined with mutually exclusive inter-cluster communications paths, through multiplexing for example, to reduce by half the number of interconnection wires required for inter-PE communications. The clustering of FIG. 5F also groups transpose elements within clusters.

Detailed Description Text (10):
One aspect of the new array's scalability is demonstrated by FIG. 5G, where a 4.times.8 torus array is depicted as two 4.times.4 arrays 40A and 40B. One could use the techniques described to this point to produce eight four-PE clusters from a 4.times.8 torus array. In addition, by dividing the 4.times.8 torus into two 4.times.4 toruses and combining respective clusters into clusters, that is clusters 44A and 44B, 46A and 46B, and so on, for example, four eight-PE clusters with all the connectivity and transpose relationships of the 4.times.4 subclusters contained in the eight four-PE cluster configuration is obtained. This cluster combining approach is general and other scalings are possible.

Detailed Description Text (11):
The presently preferred, but not sole, clustering process may also be described as follows. Given an N.times.N basic torus PE.sub.i,j, where i=0,1,2, . . . N-1 and j=0, 1, 2, . . . N-1, the preferred, South- and East-communicating clusters may be formed by grouping PE.sub.i,j, PE.sub.(i+1)(ModN), .sub.(j+N-1)(ModN), PE.sub.(i+2)(ModN), .sub.(j+N-2)(ModN), . . . , PE.sub.(i+N-1)(ModN), .sub.(j+N-(N-1))(ModN). This formula can be rewritten for an N.times.N torus array with N clusters of N PEs in which the cluster groupings can be formed by selecting an i and a j, and then using the formula: PE.sub.(i+a)(ModN), .sub.(j+N-a)(ModN) for any i,j and for all a .epsilon. {0,1, . . . , N-1}.

Detailed Description Text (12):
FIG. 6 illustrates the production of clusters 44 through 50 beginning with PE.sub.1, 3 and combining PEs which communicate to the South and East. In fact, the clusters 44 through 50, which are the clusters of the preferred embodiment of a 4.times.4 torus equivalent of the new array, are obtained by combining South and East communicating PEs, regardless of what PE within the basic N.times.N torus 40 is used as a starting point. FIGS. 7 and 8 illustrate additional examples of the approach, using 3.times.3 and 3.times.5 toruses, respectively.

Detailed Description Text (13):
Another, equivalent way of viewing the cluster-building process is illustrated in FIG. 9. In this and similar figures that follow, wraparound wires are omitted from the figure for the sake of clarity. A conventional 4.times.4 torus is first twisted into a rhombus, as illustrated by the leftward shift of each row. This shift serves to group transpose PEs in "vertical slices" of the rhombus. To produce equal-size clusters the rhombus is, basically, formed into a cylinder. That is, the left-most, or western-most, vertical slice 80 is wrapped around to abut the eastern-most PE.sub.0,3 in its row. The vertical slice 82 to the east of slice 80 is wrapped around to abut PE.sub.0,0 and PE.sub.1,3, and the next eastward vertical slice 84 is wrapped around to abut PE.sub.0,1, PE.sub.1,0 and PE.sub.2,3. Although, for the sake of clarity, all connections are not shown, all connections remain the same as in the original 4.times.4 torus. The resulting vertical slices produce the clusters of the preferred embodiment 44 through 50 shown in FIG. 5A, the same clusters produced in the manner illustrated in the discussion related to FIGS. 5A and 6. In FIG. 10, the clusters created in the rhombus/cylinder process of FIG. 9 are "peeled open" for illustrative purposes to reveal the inter-cluster connections. For example, all inter-PE connections from cluster 44 to cluster 46 are to the South and East, as are those from cluster 46 to cluster 48 and from cluster 48 to cluster 50 and from cluster 50 to cluster 44. This commonality of inter-cluster communications, in combination with the nature of inter-PE communications in a SIMD process permits a significant reduction in the number of inter-PE connections. As discussed in greater

detail in relation to FIGS. 16 and 17 below, mutually exclusive communications, e.g., communications to the South and East from cluster 44 to cluster 46 may be multiplexed onto a common set of interconnection wires running between the clusters. Consequently, the inter-PE connection wiring of the new array, hereinafter referred to as the "manifold array", may be substantially reduced, to one half the number of interconnection wires associated with a conventional nearest neighbor torus array.

Detailed Description Text (14):
The cluster formation process used to produce a manifold array is symmetrical and the clusters formed by taking horizontal slices of a vertically shifted torus are the same as clusters formed by taking vertical slices of a horizontally shifted torus. FIGS. 11A and 11B illustrate the fact that the rhombus/cylinder technique may also be employed to produce the preferred clusters from horizontal slices of a vertically shifted torus. In FIG. 11A the columns of a conventional 4.times.4 torus array are shifted vertically to produce a rhombus and in FIG. 11B the rhombus is wrapped into a cylinder. Horizontal slices of the resulting cylinder provide the preferred clusters 44 through 50. Any of the techniques illustrated to this point may be employed to create clusters for manifold arrays which provide inter-PE connectivity equivalent to that of a conventional torus array, with substantially reduced inter-PE wiring requirements.

Detailed Description Text (15):
As noted in the summary, the above clustering process is general and may be employed to produce manifold arrays of M clusters containing N PEs each from an N.times.M torus array. For example, the rhombus/cylinder approach to creating four clusters of five PEs, for a 5.times.4 torus array equivalent is illustrated in FIG. 12. Note that the vertical slices which form the new PE clusters, for example, PE.sub.4,0, PE.sub.3,1, PE.sub.2,2, PE.sub.1,3, and PE.sub.0,0 maintain the transpose clustering relationship of the previously illustrated 4.times.4 array. Similarly, as illustrated in the diagram of FIG. 13, a 4.times.5 torus will yield five clusters of four PEs each with the transpose relationship only slightly modified from that obtained with a 4.times.4 torus. In fact, transpose PEs are still clustered together, only in a slightly different arrangement than with the 4.times.4 clustered array. For example, transpose pairs PE.sub.1,0 /PE.sub.0,1 and PE.sub.2,3 /PE.sub.3,2 were grouped in the same cluster within the preferred 4.times.4 manifold array, but they appear, still paired, but in separate clusters in the 4.times.5 manifold array of FIG. 13. As illustrated in the cluster-selection diagram of FIG. 14, the diagonal PEs, PE.sub.i,j where i=j, in an odd number by odd number array are distributed one per, cluster.

Detailed Description Text (16):
The block diagrams of FIGS. 15A-15D illustrate the inter-cluster connections of the new manifold array. To simplify the description, in the following discussion, unidirectional connection paths are assumed unless otherwise stated. Although, for the sake of clarity, the invention is described with parallel interconnection paths, or buses, represented by individual lines. Bit-serial communications, in other words buses having a single line, are also contemplated by the invention. Where bus multiplexers or bus switches are used, the multiplexer and/or switches are replicated for the number of lines in the bus. Additionally, with appropriate network connections and microprocessor chip implementations of PEs, the new array may be employed with systems which allow dynamic switching between MIMD, SIMD and SISD modes, as described in U.S. Pat. No. 5,475,856 to P. M. Kogge, entitled, Dynamic Multi-Mode Parallel Processor Array Architecture, which is hereby incorporated by reference.

Detailed Description Text (17):
In FIG. 15A, clusters 80, 82 and 84 are three PE clusters connected through cluster switches 86 and inter-cluster links 88 to one another. To understand how the manifold array PEs connect to one another to create a particular topology, the connection view from a PE must be changed from that of a single PE to that of the PE as a member of a cluster of PEs. For a manifold array operating in a SIMD unidirectional communication environment, any PE requires only one transmit port and one receive port, independent of the number of connections between the PE and any of its directly attached neighborhood of PEs in the conventional torus. In general, for array communication patterns that cause no conflicts between communicating PEs, only

one transmit and one receive port are required per PE, independent of the number of neighborhood connections a particular topology may require of its PEs.

Detailed Description Text (18):
Four clusters, 44 through 50, of four PEs each are combined in the array of FIG. 15B. Cluster switches 86 and communication paths 88 connect the clusters in a manner explained in greater detail in the discussion of FIGS. 16, 17, and 18 below. Similarly, five clusters, 90 through 98, of five PEs each are combined in the array of FIG. 15C. In practice, the clusters 90-98 are placed as appropriate to ease integrated circuit layout and to reduce the length of the longest inter-cluster connection. FIG. 15D illustrates a manifold array of six clusters, 99, 100, 101, 102, 104, and 106, having six PEs each. Since communication paths 86 in the new manifold array are between clusters, the wraparound connection problem of the conventional torus array is eliminated. That is, no matter how large the array becomes, no interconnection path need be longer than the basic inter-cluster spacing illustrated by the connection paths 88. This is in contrast to wraparound connections of conventional torus arrays which must span the entire array.

Detailed Description Text (19):
The block diagram of FIG. 16 illustrates in greater detail a preferred embodiment of a four cluster, sixteen PE, manifold array. The clusters 44 through 50 are arranged, much as they would be in an integrated circuit layout, in a rectangle or square. The connection paths 88 and cluster switches are illustrated in greater detail in this figure. Connections to the South and East are multiplexed through the cluster switches 86 in order to reduce the number of connection lines between PEs. For example, the South connection between PE.sub.1,2 and PE.sub.2,2 is carried over a connection path 110, as is the East connection from PE.sub.2,1 to PE.sub.2,2. As noted above, each connection path, such as the connection path 110 may be a bit-serial path and, consequently, may be effected in an integrated circuit implementation by a single metallization line. Additionally, the connection paths are only enabled when the respective control line is asserted. These control lines can be generated by the instruction decoder/controller 38 of each PE.sub.3,0, illustrated in FIG. 3A. Alternatively, these control lines can be generated by an independent instruction decoder/controller that is included in each cluster switch. Since there are multiple PEs per switch, the multiple enable signals generated by each PE are compared to make sure they have the same value in order to ensure that no error has occurred and that all PEs are operating synchronously. That is, there is a control line associated with each noted direction path, N for North, S for South, E for East, and W for West. The signals on these lines enable the multiplexer to pass data on the associated data path through the multiplexer to the connected PE. When the control signals are not asserted the associated data paths are not enabled and data is not transferred along those paths through the multiplexer.

Detailed Description Text (20):
The block diagram of FIG. 17 illustrates in greater detail the interconnection paths 88 and switch clusters 86 which link the four clusters 44 through 50. In this figure, the West and North connections are added to the East and South connections illustrated in FIG. 16. Although, in this view, each processing element appears to have two input and two output ports, in the preferred embodiment another layer of multiplexing within the cluster switches brings the number of communications ports for each PE down to one for input and one for output. In a standard torus with four neighborhood transmit connections per PE and with unidirectional communications, that is, only one transmit direction enabled per PE, there are four multiplexer or gated circuit transmit paths required in each PE. A gated circuit may suitably include multiplexers, AND gates, tristate driver/receivers with enable and disable control signals, and other such interface enabling/disabling circuitry. This is due to the interconnection topology defined as part of the PE. The net result is that there are 4N.sup.2 multiple transmit paths in the standard torus. In the manifold array, with equivalent connectivity and unlimited communications, only 2N.sup.2 multiplexed or gated circuit transmit paths are required. This reduction of 2N.sup.2 transmit paths translates into a significant savings in integrated circuit real estate area, as the area consumed by the multiplexers and 2N.sup.2 transmit paths is significantly less than that consumed by 4N.sup.2 transmit paths.

Detailed Description Text (21):

A complete cluster switch 86 is illustrated in greater detail in the block diagram of FIG. 18. The North, South, East, and West outputs are as previously illustrated. Another layer of multiplexing 112 has been added to the cluster switch 86. This layer of multiplexing selects between East/South reception, labeled A, and North/West reception, labeled B, thereby reducing the communications port requirements of each PE to one receive port and one send port. Additionally, multiplexed connections between transpose PEs, PE.sub.1,3 and PE.sub.3,1, are effected through the intra-cluster transpose connections labeled T. When the T multiplexer enable signal for a particular multiplexer is asserted, communications from a transpose PE are received at the PE associated with the multiplexer. In the preferred embodiment, all clusters include transpose paths such as this between a PE and its transpose PE. These figures illustrate the overall connection scheme and are not intended to illustrate how a multi-layer integrated circuit implementation may accomplish the entirety of the routine array interconnections that would typically be made as a routine matter of design choice. As with any integrated circuit layout, the IC designer would analyze various tradeoffs in the process of laying out an actual IC implementation of an array in accordance with the present invention. For example, the cluster switch may be distributed within the PE cluster to reduce the wiring lengths of the numerous interfaces.

Detailed Description Text (22):
To demonstrate the equivalence to a torus array's communication capabilities and the ability to execute an image processing algorithm on the Manifold Array, a simple 2D convolution using a 3.times.3 window, FIG. 19A, will be described below. The Lee and Aggarwal algorithm for convolution on a torus machine will be used. See, S. Y. Lee and J. K. Aggarwal, Parallel 2D Convolution on a Mesh Connected Array Processor, IEEE Transactions on Patter Analysis and Machine Intelligence, Vol. PAMI-9, No. 4, pp. 590-594, July 1987. The internal structure of a basic PE 30, FIG. 3A, is used to demonstrate the convolution as executed on a 4.times.4 Manifold Array with 16 of these PEs. For purposes of this example, the Instruction Decoder/Controller also provides the Cluster Switch multiplexer Enable signals. Since there are multiple PEs per switch, the multiple enable signals are compared to be equal to ensure no error has occurred and all PEs are operating in synchronism. Based upon the S. Y. Lee and J. K. Aggarwal algorithm for convolution, the Manifold array would desirably be the size of the image, for example, an N.times.N array for a N.times.N image. Due to implementation issues it must be assumed that the array is smaller than N.times.N for large N. Assuming the array size is C.times.C, the image processing can be partitioned into multiple C.times.C blocks, taking into account the image block overlap required by the convolution window size. Various techniques can be used to handle the edge effects of the N.times.N image. For example, pixel replication can be used that effectively generates an (N+1).times.(N+1) array. It is noted that due to the simplicity of the processing required, a very small PE could be defined in an application specific implementation. Consequently, a large number of PEs could be placed in a Manifold Array organization on a chip thereby improving the efficiency of the convolution calculations for large image sizes.

Detailed Description Text (23):
The convolution algorithm provides a simple means to demonstrate the functional equivalence of the Manifold Array organization to a torus array for North/East/South/West nearest neighbor communication operations. Consequently, the example focuses on the communications aspects of the algorithm and, for simplicity of discussion, a very small 4.times.4 image size is used on a 4.times.4 Manifold array. Larger N.times.N images can be handled in this approach by loading a new 4.times.4 image segment into the array after each previous 4.times.4 block is finished. For the 4.times.4 array no wrap around is used and for the edge PEs 0's are received from the virtual PEs not present in the physical implementation. The processing for one 4.times.4 block of pixels will be covered in this operating example.

Detailed Description Text (24):
To begin the convolution example, it is assumed that the PEs have already been initialized by a SIMD controller, such as controller 29 of FIG. 3A, and the initial 4.times.4 block of pixels has been loaded through the data bus to register R1 in each PE, in other words, one pixel per PE has been loaded. FIG. 19C shows a portion of an image with a 4.times.4 block to be loaded into the array. FIG. 19D shows this

block loaded in the 4.times.4 torus logical positions. In addition, it is assumed that the accumulating sum register R0 in each PE has been initialized to zero. Though inconsequential to this algorithm, R2 has also been shown as initialized to zero. The convolution window elements are broadcast one at a time in each step of the algorithm. These window elements are received into register R2. The initial state of the machine prior to broadcasting the window elements is shown in FIG. 20A. The steps to calculate the sum of the weighted pixel values in a 3.times.3 neighborhood for all PEs follows.

Detailed Description Text (38):
The above example demonstrates that the Manifold Array is equivalent in its communications capabilities for the four--North, East, South, and West--communications directions of a standard torus while requiring only half the wiring expense of the standard torus. Given the Manifold Array's capability to communicate between transpose PEs, implemented with a regular connection pattern, minimum wire length, and minimum cost, the Manifold Array provides additional capabilities beyond the standard torus. Since the Manifold Array organization is more regular as it is made up of the same size clusters of PEs while still providing the communications capabilities of transpose and neighborhood communications, it represents a superior design to the standard and diagonal fold toruses of the prior art.

Other Reference Publication (1):
Pechanek et al. "Multiple-Fold Clustered Processor Mesh Array", Proceedings Fifth NASA Symposium on VLSI Design, pp. 8.4.1-11, Nov. 4-5-1993,University of New Mexico, Albuquerque, New Mexico.

CLAIMS:

1. An interconnection system for connecting a plurality of processing elements (PEs) in a torus-connected PE array, each PE having a communications port for communicating with the other PEs, the communications port including a single input and a single output, the interconnection system comprising:

inter-PE connection paths for connecting PEs grouped in clusters through cluster switches, with each cluster of PEs communicating with two other clusters of PEs in mutually exclusive directions through the cluster switches and inter-PE connection paths; and

the cluster switches connected to both the communications ports of said PEs and the inter-PE connection paths, and controllably switched to multiplex mutually exclusive communications onto the inter-PE connection paths connecting the cluster switches to reduce the number of communications paths required to provide inter-PE connectivity.

2. The interconnection system of claim 1, wherein a predetermined number of said plurality of PEs form pairs of transpose PEs, and wherein said cluster switches further comprise intra-cluster transpose connections to provide direct communications between the pairs of transpose PEs.

3. The interconnection system of claim 1, further comprising a control connected to the cluster switches for controlling the controllably switched cluster switches to select selectable modes of operation and wherein data and commands may be transmitted and received at said communications ports in one of four selectable modes:

a) a transmit east/receive west mode for transmitting data to an east PE via the communications port of the east PE while receiving data from a west PE via the communications port of the west PE;

b) a transmit north/receive south mode for transmitting data to a north PE via the communications port of the north PE while receiving data from a south PE via the communications port of the south PE;

c) a transmit south/receive north mode for transmitting data to an south PE via the

communications port of the south PE while receiving data from a north PE via the communications port of the north PE; and

d) a transmit west/receive east mode for transmitting data to a west PE via the communications port of the west PE while receiving data from an east PE via the communications port of the east PE.

6. The interconnection system of claim 5, wherein said inter-PE connection paths are selectively switched through the cluster switches to select between different connection paths by paths enabling signals.

11. The interconnection system of claim 9, wherein the cluster switch supports an operation wherein the PEs are each simultaneously sending commands or data through the output while receiving commands or data through the input.

13. An array processor, comprising:

a plurality of processing elements (PEs) grouped in clusters, with each cluster communicating with two other clusters in mutually exclusive directions, each PE having a single inter-PE communications port for communicating with other PEs, each of said ports having a single input and a single output;

inter-PE communications paths connecting said single inter-PE communications ports through controllably switched cluster switches; and

the controllably switched cluster switches to select mutually exclusive inter-PE connection paths for PE to PE communication and connect the plurality of PEs into a torus connected array.

15. An array processor, comprising:

a plurality of processing elements (PEs) arranged in clusters, each each PE having a communications port for communicating with the other PEs, the communications port including a single input and a single output;

inter-PE communications paths connecting the PEs through cluster switches; and

the cluster switches operable to multiplex inter-PE communications and connect the PEs of each cluster for communication in mutually exclusive directions with the PEs of each of at least two other clusters utilizing the inter-PE communication paths.

Full | Title | CLS.1 REF.1 SEQ.1 ATT.1

Generate Collection | Print

Term	Documents
TORUS	4936
TORU	7583
(3 AND TORUS).USPT.	1
(L3 AND (TORUS)).USPT.	1

Display Format: KWIC Change Format

Previous Page Next Page

Set Name side by side	**Query**	**Hit Count**	**Set Name** result set
	DB=USPT; PLUR=YES; OP=ADJ		
L15	L13 and L14	5	L15
L14	L8 and ((processing adj1 element$) with (torus))	15	L14
L13	L8 and ((processing adj1 element$) with (cluster$))	45	L13
L12	L11 and (cluster$)	133	L12
L11	L8 and (processing adj1 element$)	787	L11
L10	L9 and (processing adj1 element$)	94	L10
L9	L8 and (torus)	176	L9
L8	(712/$ OR 364/$)!.CCLS.	8232	L8
L7	(712/$ OR 364/$).CCLS.	8232	L7
L6	(712/$ or 364/$).ccls	0	L6
L5	L3 and (torus)	1	L5
L4	L3 and (torus adj1 direction$)	0	L4
L3	L2 and (cluster adj1 switch$)	1	L3
L2	L1 and cluster$	1	L2
L1	6338129.pn.	1	L1

END OF SEARCH HISTORY

WEST

Generate Collection

L15: Entry 5 of 5 File: USPT Feb 10, 1998

DOCUMENT-IDENTIFIER: US 5717943 A
TITLE: Advanced parallel array processor (APAP)

Detailed Description Text (56):
FIG. 6 illustrates the fine-grained parallel technology path from the single processor element 300, made up of 32K 16-bit words with a 16-bit processor to the Network node 310 of eight processors 312 and their associated memory 311 with their fully distributed I/O routers 313 and Signal I/O ports 314, 315, on through groups of nodes labeled clusters 320 and into the cluster configuration 360 and to the various applications 330, 340, 350, 370. The 2d level structure is the cluster 320, and 64 clusters are integrated to form the 4d modified hypercube of 32,768 Processing Elements 360.

Detailed Description Text (305):
The packaging concept is intended to significantly reduce the off page wire count for each of the clusters. This concept takes a cluster which is defined as a 8.times.8 array of nodes 820, each node 825 having 8 processing elements for a total of 512 PMEs, then to limit the X and Y ring within the cluster and, finally, to bring out the W and Z buses to all clusters. The physical picture could be envisioned as a sphere configuration 800, 810 of 64 smaller spheres 830. See FIG. 15 for a future packaging picture which illustrates the full up packaging technique, limiting the X and Y rings 800 within the cluster and extending out the W and Z buses to all clusters 810. The physical picture could be envisioned as a sphere configuration of 64 smaller spheres 830.

Current US Original Classification (1):
712/20

Current US Cross Reference Classification (1):
712/14

CLAIMS:

113. A multi-processor memory system comprising: on a chip a plurality of processing elements with a network interface, said processing elements of said chip being intercoupled by an internal communication network for passing information between processing elements on the chip, and having a broadcast port for external communication from the chip, said processing elements on the chip have their own memory and they are coupled in a network as a torus, said system having a plurality of chips which are coupled chip to chip to form a parallel array of multiple nodes of chips, each node having a broadcast and control interface for communications between processing elements on a chip and between nodes.

117. A parallel array computer system, comprising: a plurality of processing elements each having accessible memory and organized as cluster of processing elements, each of said processing elements of a cluster having a fast I/O tri-state driver;

wherein the parallel array computer system provides a multi-processor memory system including a PME architecture multi-processor memory element on a single semiconductor substrate which functions as a system node, said multi-processor memory element including a plurality of processing memory elements, and means on said substrate for distributing interconnection and controls within the multi-processor memory system node enabling the system to perform SIMD/MIMD functions as a multi-processor memory system, wherein each dedicated local memory is

independently accessible by the respectively coupled processor in both SIMD and MIMD modes exclusive of access by another processor.

Set Name side by side	**Query**	**Hit Count**	**Set Name** result set
	DB=USPT; PLUR=YES; OP=ADJ		
L12	110 and (array adj processor) and (communic$ with mutually with exclusive$)	0	L12
L11	(5148515 or 5475856 or 5566342 or 5682491).pn.	4	L11
L10	(5148515 or 5475856 or 5566342 or 5682491).pn.	4	L10
L9	(array with processor$) same (communicat$ with mutually with exclusive)	5	L9
L8	(array with processor$).ab. and (communicat$ with mutually with exclusive)	10	L8
L7	L5 and (communicat$ with mutually with exclusive)	2	L7
L6	L5 and (communicat$ with mutually with exclusive with torus)	2	L6
L5	(6023753 or 6338129).pn.	2	L5
L4	6023753.pn.	1	L4
L3	6338129.pn.	1	L3
~~L2~~	~~l1 and (load and balanc$).ti.~~	~~51~~	~~L2~~
~~L1~~	~~(load and balanc$ and (network$ or server$)).ab.~~	~~186~~	~~L1~~

END OF SEARCH HISTORY

WEST

Generate Collection | Print

Search Results - Record(s) 1 through 2 of 2 returned.

☐ 1. Document ID: US 6338129 B1

L7: Entry 1 of 2 File: USPT Jan 8, 2002

DOCUMENT-IDENTIFIER: US 6338129 B1
TITLE: Manifold array processor

US PATENT NO. (1):
6338129

Abstract Text (1):
An array processor includes processing elements arranged in clusters which are, in turn, combined in a rectangular array. Each cluster is formed of processing elements which preferably communicate with the processing elements of at least two other clusters. Additionally each inter-cluster communication path is mutually exclusive, that is, each path carries either north and west, south and east, north and east, or south and west communications. Due to the mutual exclusivity of the data paths, communications between the processing elements of each cluster may be combined in a single inter-cluster path. That is, communications from a cluster which communicates to the north and east with another cluster may be combined in one path, thus eliminating half the wiring required for the path. Additionally, the length of the longest communication path is not directly determined by the overall dimension of the array, as it is in conventional torus arrays. Rather, the longest communications path is limited only by the inter-cluster spacing. In one implementation, transpose elements of an N.times.N torus are combined in clusters and communicate with one another through intra-cluster communications paths. Since transpose elements have direct connections to one another, transpose operation latency is eliminated in this approach. Additionally, each PE may have a single transmit port and a single receive port. As a result, the individual PEs are decoupled from the topology of the array.

Brief Summary Text (14):
To form an array in accordance with the present invention, processing elements may first be combined into clusters which capitalize on the communications requirements of single instruction multiple data ("SIMD") operations. Processing elements may then be grouped so that the elements of one cluster communicate within a cluster and with members of only two other clusters. Furthermore, each cluster's constituent processing elements communicate in only two mutually exclusive directions with the processing elements of each of the other clusters. By definition, in a SIMD torus with unidirectional communication capability, the North/South directions are mutually exclusive with the East/West directions. Processing element clusters are, as the name implies, groups of processors formed preferably in close physical proximity to one another. In an integrated circuit implementation, for example, the processing elements of a cluster preferably would be laid out as close to one another as possible, and preferably closer to one another than to any other processing element in the array. For example, an array corresponding to a conventional four by four torus array of processing elements may include four clusters of four elements each, with each cluster communicating only to the North and East with one other cluster and to the South and West with another cluster, or to the South and East with one other cluster and to the North and West with another cluster. By clustering PEs in this manner, communications paths between PE clusters may be shared, through multiplexing, thus substantially reducing the interconnection wiring required for the array.

Detailed Description Text (2):
In one embodiment, a new array processor in accordance with the present invention combines PEs in clusters, or groups, such that the elements of one cluster communicate with members of only two other clusters and each cluster's constituent processing elements communicate in only two mutually exclusive directions with the processing elements of each of the other clusters. By clustering PEs in this manner, communications paths between PE clusters may be shared, thus substantially reducing the interconnection wiring required for the array. Additionally, each PE may have a single transmit port and a single receive port or, in the case of a bidirectional sequential or time sliced transmit/receive communication implementation, a single transmit/receive port. As a result, the individual PEs are decoupled from the topology of the array. That is, unlike a conventional torus connected array where each PE has four bidirectional communication ports, one for communication in each direction, PEs employed by the new array architecture need only have one port. In implementations which utilize a single transmit and a single receive port, all PEs in the array may simultaneously transmit and receive. In the conventional torus, this would require four transmit and four receive ports, a total of eight ports, per PE, while in the present invention, one transmit port and one receive port, a total of two ports, per PE are required.

Detailed Description Text (7):
In FIG. 5A, the basic 4.times.4 PE torus is once again surrounded by tilings of itself. The present invention recognizes that communications to the East and South from PE.sub.0,0 involve PE.sub.0,1 and PE.sub.1,0, respectively. Furthermore, the PE which communicates to the east to PE.sub.1,0 is PE.sub.1,3 and PE.sub.1,3 communicates to the South to PE.sub.2,3. Therefore, combining the four PEs, PE.sub.0,0, PE.sub.1,3, PE.sub.2,2, and PE.sub.3,1 in one cluster yields a cluster 44 from which PEs communicate only to the South and East with another cluster 46 which includes PEs, PE.sub.0,1, PE.sub.1,0, PE.sub.2,3 and PE.sub.3,2. Similarly, the PEs of cluster 46 communicate to the South and East with the PEs of cluster 48 which includes PEs, PE.sub.0,2, PE.sub.1,1, PE.sub.2,0, and PE.sub.3,3. The PEs, PE.sub.0,3, PE.sub.1,2, PE.sub.2,1, and PE.sub.3,0 of cluster 50 communicate to the South and East with cluster 44. This combination yields clusters of PEs which communicate with PEs in only two other clusters and which communicate in mutually exclusive directions to those clusters. That is, for example, the PEs of cluster 48 communicate only to the South and East with the PEs of cluster 50 and only to the North and West with the PEs of cluster 46. It is this exemplary of grouping of PEs which permits the inter-PE connections within an array in accordance with the present invention to be substantially reduced in comparison with the requirements of the conventional nearest neighbor torus array.

Detailed Description Text (9):
Another clustering is depicted in FIG. 5F where clusters 61, 63, 65, and 67 form a criss cross pattern in the tilings of the torus 40. This clustering demonstrates that there are a number of ways in which to group PEs to yield clusters which communicate with two other clusters in mutually exclusive directions. That is, PE.sub.0,0 and PE.sub.2,2 of cluster 65 communicate to the East with PE.sub.0,1 and PE.sub.2,3, respectively, of cluster 61. Additionally, PE.sub.1,1 and PE.sub.3,3 of cluster 65 communicate to the West with PE.sub.1,0 and PE.sub.3,2, respectively, of cluster 61. As will be described in greater detail below, the Easterly communications paths just described, that is, those between PE.sub.0,0 and PE.sub.0,1 and between PE.sub.2,2 and PE.sub.2,3 and other inter-cluster paths may be combined with mutually exclusive inter-cluster communications paths, through multiplexing for example, to reduce by half the number of interconnection wires required for inter-PE communications. The clustering of FIG. 5F also groups transpose elements within clusters.

Detailed Description Text (13):
Another, equivalent way of viewing the cluster-building process is illustrated in FIG. 9. In this and similar figures that follow, wraparound wires are omitted from the figure for the sake of clarity. A conventional 4.times.4 torus is first twisted into a rhombus, as illustrated by the leftward shift of each row. This shift serves to group transpose PEs in "vertical slices" of the rhombus. To produce equal-size clusters the rhombus is, basically, formed into a cylinder. That is, the left-most, or western-most, vertical slice 80 is wrapped around to abut the eastern-most

PE.sub.0,3 in its row. The vertical slice 82 to the east of slice 80 is wrapped around to abut PE.sub.0,0 and PE.sub.1,3, and the next eastward vertical slice 84 is wrapped around to abut PE.sub.0,1, PE.sub.1,0 and PE.sub.2,3. Although, for the sake of clarity, all connections are not shown, all connections remain the same as in the original 4.times.4 torus. The resulting vertical slices produce the clusters of the preferred embodiment 44 through 50 shown in FIG. 5A, the same clusters produced in the manner illustrated in the discussion related to FIGS. 5A and 6. In FIG. 10, the clusters created in the rhombus/cylinder process of FIG. 9 are "peeled open" for illustrative purposes to reveal the inter-cluster connections. For example, all inter-PE connections from cluster 44 to cluster 46 are to the South and East, as are those from cluster 46 to cluster 48 and from cluster 48 to cluster 50 and from cluster 50 to cluster 44. This commonality of inter-cluster communications, in combination with the nature of inter-PE communications in a SIMD process permits a significant reduction in the number of inter-PE connections. As discussed in greater detail in relation to FIGS. 16 and 17 below, mutually exclusive communications, e.g., communications to the South and East from cluster 44 to cluster 46 may be multiplexed onto a common set of interconnection wires running between the clusters. Consequently, the inter-PE connection wiring of the new array, hereinafter referred to as the "manifold array", may be substantially reduced, to one half the number of interconnection wires associated with a conventional nearest neighbor torus array.

CLAIMS:

1. An interconnection system for connecting a plurality of processing elements (PEs) in a torus-connected PE array, each PE having a communications port for communicating with the other PEs, the communications port including a single input and a single output, the interconnection system comprising:

inter-PE connection paths for connecting PEs grouped in clusters through cluster switches, with each cluster of PEs communicating with two other clusters of PEs in mutually exclusive directions through the cluster switches and inter-PE connection paths; and

the cluster switches connected to both the communications ports of said PEs and the inter-PE connection paths, and controllably switched to multiplex mutually exclusive communications onto the inter-PE connection paths connecting the cluster switches to reduce the number of communications paths required to provide inter-PE connectivity.

13. An array processor, comprising:

a plurality of processing elements (PEs) grouped in clusters, with each cluster communicating with two other clusters in mutually exclusive directions, each PE having a single inter-PE communications port for communicating with other PEs, each of said ports having a single input and a single output;

inter-PE communications paths connecting said single inter-PE communications ports through controllably switched cluster switches; and

the controllably switched cluster switches to select mutually exclusive inter-PE connection paths for PE to PE communication and connect the plurality of PEs into a torus connected array.

15. An array processor, comprising:

a plurality of processing elements (PEs) arranged in clusters, each each PE having a communications port for communicating with the other PEs, the communications port including a single input and a single output;

inter-PE communications paths connecting the PEs through cluster switches; and

the cluster switches operable to multiplex inter-PE communications and connect the PEs of each cluster for communication in mutually exclusive directions with the PEs of each of at least two other clusters utilizing the inter-PE communication paths.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 2. Document ID: US 6023753 A

L7: Entry 2 of 2 File: USPT Feb 8, 2000

DOCUMENT-IDENTIFIER: US 6023753 A
TITLE: Manifold array processor

US PATENT NO. (1):
6023753

Abstract Text (1):
An array processor includes processing elements arranged in clusters which are, in turn, combined in a rectangular array. Each cluster is formed of processing elements which preferably communicate with the processing elements of at least two other clusters. Additionally each inter-cluster communication path is mutually exclusive, that is, each path carries either north and west, south and east, north and east, or south and west communications. Due to the mutual exclusivity of the data paths, communications between the processing elements of each cluster may be combined in a single inter-cluster path. That is, communications from a cluster which communicates to the north and east with another cluster may be combined in one path, thus eliminating half the wiring required for the path. Additionally, the length of the longest communication path is not directly determined by the overall dimension of the array, as it is in conventional torus arrays. Rather, the longest communications path is limited only by the inter-cluster spacing. In one implementation, transpose elements of an N.times.N torus are combined in clusters and communicate with one another through intra-cluster communications paths. Since transpose elements have direct connections to one another, transpose operation latency is eliminated in this approach. Additionally, each PE may have a single transmit port and a single receive port. As a result, the individual PEs are decoupled from the topology of the array.

Brief Summary Text (15):
To form an array in accordance with the present invention, processing elements may first be combined into clusters which capitalize on the communications requirements of single instruction multiple data ("SIMD") operations. Processing elements may then be grouped so that the elements of one cluster communicate within a cluster and with members of only two other clusters. Furthermore, each cluster's constituent processing elements communicate in only two mutually exclusive directions with the processing elements of each of the other clusters. By definition, in a SIMD torus with unidirectional communication capability, the North/South directions are mutually exclusive with the East/West directions. Processing element clusters are, as the name implies, groups of processors formed preferably in close physical proximity to one another. In an integrated circuit implementation, for example, the processing elements of a cluster preferably would be laid out as close to one another as possible, and preferably closer to one another than to any other processing element in the array. For example, an array corresponding to a conventional four by four torus array of processing elements may include four clusters of four elements each, with each cluster communicating only to the North and East with one other cluster and to the South and West with another cluster, or to the South and East with one other cluster and to the North and West with another cluster. By clustering PEs in this manner, communications paths between PE clusters may be shared, through multiplexing, thus substantially reducing the interconnection wiring required for the array.

Detailed Description Text (2):
In one embodiment, a new array processor in accordance with the present invention combines PEs in clusters, or groups, such that the elements of one cluster communicate with members of only two other clusters and each cluster's constituent

processing elements communicate in only two mutually exclusive directions with the processing elements of each of the other clusters. By clustering PEs in this manner, communications paths between PE clusters may be shared, thus substantially reducing the interconnection wiring required for the array. Additionally, each PE may have a single transmit port and a single receive port or, in the case of a bidirectional sequential or time sliced transmit/receive communication implementation, a single transmit/receive port. As a result, the individual PEs are decoupled from the topology of the array. That is, unlike a conventional torus connected array where each PE has four bidirectional communication ports, one for communication in each direction, PEs employed by the new array architecture need only have one port. In implementations which utilize a single transmit and a single receive port, all PEs in the array may simultaneously transmit and receive. In the conventional torus, this would require four transmit and four receive ports, a total of eight ports, per PE, while in the present invention, one transmit port and one receive port, a total of two ports, per PE are required.

Detailed Description Text (8):
In FIG. 5A, the basic 4.times.4 PE torus is once again surrounded by tilings of itself. The present invention recognizes that communications to the East and South from PE.sub.0,0 involve PE.sub..sub.0,1 and PE.sub.1,0, respectively. Furthermore, the PE which communicates to the east to PE.sub.1,0 is PE.sub.1,3 and PE.sub.1,3 communicates to the South to PE.sub.2,3. Therefore, combining the four PEs, PE.sub.0,0, PE.sub.1,3, PE.sub.2,2, and PE.sub.3,1 in one cluster yields a cluster 44 from which PEs communicate only to the South and East with another cluster 46 which includes PEs, PE.sub.0,1, PE.sub.1,0, PE.sub.2,3 and PE.sub.3,2. Similarly, the PEs of cluster 46 communicate to the South and East with the PEs of cluster 48 which includes PEs, PE.sub.0,2, PE.sub.1,1, PE.sub.2,0, and PE.sub.3,3. The PEs, PE.sub.0,3, PE.sub.1,2, PE.sub.2,1, and PE.sub.3,0 of cluster 50 communicate to the South and East with cluster 44. This combination yields clusters of PEs which communicate with PEs in only two other clusters and which communicate in mutually exclusive directions to those clusters. That is, for example, the PEs of cluster 48 communicate only to the South and East with the PEs of cluster 50 and only to the North and West with the PEs of cluster 46. It is this exemplary of grouping of PEs which permits the inter-PE connections within an array in accordance with the present invention to be substantially reduced in comparison with the requirements of the conventional nearest neighbor torus array.

Detailed Description Text (10):
Another clustering is depicted in FIG. 5F where clusters 61, 63, 65, and 67 form a criss cross pattern in the tilings of the torus 40. This clustering demonstrates that there are a number of ways in which to group PEs to yield clusters which communicate with two other clusters in mutually exclusive directions. That is, PE.sub.0,0 and PE.sub.2,2 of cluster 65 communicate to the East with PE.sub.0,1 and PE.sub.2,3, respectively, of cluster 61. Additionally, PE.sub.1,1 and PE.sub.3,3 of cluster 65 communicate to the West with PE.sub.1,0 and PE.sub.3,2, respectively, of cluster 61. As will be described in greater detail below, the Easterly communications paths just described, that is, those between PE.sub.0,0 and PE.sub.0,1 and between PE.sub.2,2 and PE.sub.2,3, and other inter-cluster paths may be combined with mutually exclusive inter-cluster communications paths, through multiplexing for example, to reduce by half the number of interconnection wires required for inter-PE communications. The clustering of FIG. 5F also groups transpose elements within clusters.

Detailed Description Text (14):
Another, equivalent way of viewing the cluster-building process is illustrated in FIG. 9. In this and similar figures that follow, wraparound wires are omitted from the figure for the sake of clarity. A conventional 4.times.4 torus is first twisted into a rhombus, as illustrated by the leftward shift of each row. This shift serves to group transpose PEs in "vertical slices" of the rhombus. To produce equal-size clusters the rhombus is, basically, formed into a cylinder. That is, the left-most, or western-most, vertical slice 80 is wrapped around to abut the eastern-most PE.sub.0,3 in its row. The vertical slice 82 to the east of slice 80 is wrapped around to abut PE.sub.0,0 and PE.sub.1,3, and the next eastward vertical slice 84 is wrapped around to abut PE.sub.0,1, PE.sub.1,0 and PE.sub.2,3. Although, for the sake of clarity, all connections are not shown, all connections remain the same as in the

original 4.times.4 torus. The resulting vertical slices produce the clusters of the preferred embodiment 44 through 50 shown in FIG. 5A, the same clusters produced in the manner illustrated in the discussion related to FIGS. 5A and 6. In FIG. 10, the clusters created in the rhombus/cylinder process of FIG. 9 are "peeled open" for illustrative purposes to reveal the inter-cluster connections. For example, all inter-PE connections from cluster 44 to cluster 46 are to the South and East, as are those from cluster 46 to cluster 48 and from cluster 48 to cluster 50 and from cluster 50 to cluster 44. This commonality of inter-cluster communications, in combination with the nature of inter-PE communications in a SIMD process permits a significant reduction in the number of inter-PE connections. As discussed in greater detail in relation to FIGS. 16 and 17 below, mutually exclusive communications, e.g., communications to the South and East from cluster 44 to cluster 46 may be multiplexed onto a common set of interconnection wires running between the clusters. Consequently, the inter-PE connection wiring of the new array, hereinafter referred to as the "manifold array", may be substantially reduced, to one half the number of interconnection wires associated with a conventional nearest neighbor torus array.

CLAIMS:

1. An array processor, comprising:

N clusters wherein each cluster contains M processing elements, each processing element having a communications port through which the processing element transmits and receives data over a total of B wires;

communications paths which are less than or equal to (M)(B)-wires wide connected between pairs of said clusters; each cluster member in the pair containing processing elements which are torus nearest neighbors to processing elements in the other cluster of the pair, each path permitting communications between said cluster pairs in two mutually exclusive torus directions, that is, South and East or South and West or North and East or North and West; and

multiplexers connected to combine 2(M)(B)-wire wide communications into said less than or equal to (M)(B)-wires wide paths between said cluster pairs.

5. The array processor of claim 1, wherein a cluster switch comprises said multiplexers and said cluster switch is connected to mutliplex communications received from two mutually exclusive torus directions to processing elements within a cluster.

10. An array processor, comprising:

N clusters wherein each cluster contains M processing elements, each processing element having a communications port through which the processing element transmits and receives data over a total of B wires and each processing element within a cluster being formed in closer physical proximity to other processing elements within a cluster than to processing elements outside the cluster;

communications paths which are less than or equal to (M)(B)-wires wide connected between pairs of said clusters, each cluster member in the pair containing processing elements which are torus nearest neighbors to processing elements in the other cluster of the pair, each path permitting communications between said cluster pairs in two mutually exclusive torus directions, that is, South and East or South and West or North and East or North and West; and

multiplexers connected to combine 2(M)(B)-wire wide communications into said less than or equal to (M)(B)-wires wide paths between said cluster pairs.

14. The array processor of claim 10, wherein a cluster switch comprises said multiplexer and said cluster switch is connected to mutliplex communications received from two mutually exclusive torus directions to processing elements within a cluster.

28. A method of forming an array processor, comprising the steps of:

arranging processing elements in N clusters wherein each cluster contains M processing elements, such that each cluster includes processing elements which communicate only in mutually exclusive torus directions with the processing elements of at least one other cluster; and

multiplexing said mutually exclusive torus direction communications.

Generate Collection | Print

Term	Documents
MUTUALLY.USPT.	140445
MUTUALLIES	0
MUTUALLYS	0
EXCLUSIVE.USPT.	82794
EXCLUSIVES.USPT.	24
COMMUNICAT$	0
COMMUNICAT.USPT.	10
COMMUNICATABILITY.USPT.	3
COMMUNICATABLE.USPT.	500
COMMUNICATABLY.USPT.	39
COMMUNICATAED.USPT.	1
(L5 AND (COMMUNICAT$ WITH MUTUALLY WITH EXCLUSIVE)).USPT.	2

There are more results than shown above. Click here to view the entire set.

Display Format: KWIC | Change Format

Previous Page Next Page

WEST

Generate Collection

L9: Entry 1 of 5 File: USPT Oct 22, 2002

DOCUMENT-IDENTIFIER: US 6470441 B1
TITLE: Methods and apparatus for manifold array processing

Detailed Description Text (2):
In one embodiment, a manifold array processor in accordance with the present invention combines PEs in clusters, or groups, such that the elements of one cluster communicate directly with members of only two other clusters and each cluster's constituent processing elements communicate directly in only two mutually exclusive directions with the processing elements of each of the other clusters. By clustering PEs in this manner, communications paths between PE clusters may be shared, thus substantially reducing the interconnection wiring required for an array. Additionally, each PE may have a single transmit port and a single receive port or, in the case of a bidirectional, sequential or time-sliced communications implementation, a single transmit/receive port. As a result, the individual PE are de-coupled from the array architecture. That is, unlike a conventional N-dimensional hypercube-connected array where each PE has N communication ports. In implementations which utilize a single transmit and a single receive port, all PEs in the array may simultaneously transmit and receive. In a conventional 6D hypercube, this would require six transmit and six receive ports, a total of twelve data ports, for each PE. With the present invention, only one transmit- and one receive-port, a total of two data ports are required, regardless of the hypercube's dimension. As noted above, the transmit and receive data ports may be combined into one transmit/receive data port if bidirectional, sequential or time-sliced data communications are employed. Each PE contains a virtual PE storage unit and a configuration control unit. The virtual PE number and configuration control information are combined to determine the settings of cluster switches, to control the direction of communications, and to reconfigure the PE array's topology. This reconfiguration may be in response to a dispatched instruction from a controller, for example. PEs within an array are clustered so that a PE and its transpose are combined within a cluster and a PE and its hypercube complement are contained with the same cluster.

First Hit Fwd Refs

Generate Collection

L2: Entry 3 of 11 File: USPT May 28, 2002

DOCUMENT-IDENTIFIER: US 6397324 B1
TITLE: Accessing tables in memory banks using load and store address generators sharing store read port of compute register file separated from address register file

Detailed Description Text (6):
Interconnecting the PEs for data transfer communications is the cluster switch 171 more completely described in U.S. Pat. No. 6,023,753 entitled "Manifold Array Processor", U.S. application Ser. No. 09/949,122 entitled "Methods and Apparatus for Manifold Array Processing", and U.S. application Ser. No. 09/169,256 entitled "Methods and Apparatus for ManArray PE-to-PE Switch Control". The interface to a host processor, other peripheral devices, and/or external memory can be done in many ways. The primary mechanism shown for completeness is contained in a direct memory access (DMA) control unit 181 that provides a scalable ManArray data bus 183 that connects to devices and interface units external to the ManArray core. The DMA control unit 181 provides the data flow and bus arbitration mechanisms needed for these external devices to interface to the ManArray core memories via the multiplexed bus interface represented by line 185. A high level view of a ManArray Control Bus (MCB) 191 is also shown.

CLAIMS:

5. The apparatus of claim 1 wherein said processor is an array controller sequence processor.

WEST

Generate Collection

L9: Entry 3 of 5 File: USPT Dec 26, 2000

DOCUMENT-IDENTIFIER: US 6167502 A
TITLE: Method and apparatus for manifold array processing

Detailed Description Text (2):
In one embodiment, a manifold array processor in accordance with the present invention combines PEs in clusters, or groups, such that the elements of one cluster communicate directly with members of only two other clusters and each cluster's constituent processing elements communicate directly in only two mutually exclusive directions with the processing elements of each of the other clusters. By clustering PEs in this manner, communications paths between PE clusters may be shared, thus substantially reducing the interconnection wiring required for an array. Additionally, each PE may have a single transmit port and a single receive port or, in the case of a bidirectional, sequential or time-sliced communications implementation, a single transmit/receive port. As a result, the individual PE are de-coupled from the array architecture. That is, unlike a conventional N-dimensional hypercube-connected array where each PE has N communication ports. In implementations which utilize a single transmit and a single receive port, all PEs in the array may simultaneously transmit and receive. In a conventional 6D hypercube, this would require six transmit and six receive ports, a total of twelve data ports, for each PE. With the present invention, only one transmit- and one receive-port, a total of two data ports are required, regardless of the hypercube's dimension. As noted above, the transmit and receive data ports may be combined into one transmit/receive data port if bidirectional, sequential or time-sliced data communications are employed. Each PE contains a virtual PE storage unit and a configuration control unit. The virtual PE number and configuration control information are combined to determine the settings of cluster switches, to control the direction of communications, and to reconfigure the PE array's topology. This reconfiguration may be in response to a dispatched instruction from a controller, for example. PEs within an array are clustered so that a PE and its transpose are combined within a cluster and a PE and its hypercube complement are contained with the same cluster.

WEST

Generate Collection

L9: Entry 4 of 5 File: USPT Feb 8, 2000

DOCUMENT-IDENTIFIER: US 6023753 A
TITLE: Manifold array processor

Abstract Text (1):
An array processor includes processing elements arranged in clusters which are, in turn, combined in a rectangular array. Each cluster is formed of processing elements which preferably communicate with the processing elements of at least two other clusters. Additionally each inter-cluster communication path is mutually exclusive, that is, each path carries either north and west, south and east, north and east, or south and west communications. Due to the mutual exclusivity of the data paths, communications between the processing elements of each cluster may be combined in a single inter-cluster path. That is, communications from a cluster which communicates to the north and east with another cluster may be combined in one path, thus eliminating half the wiring required for the path. Additionally, the length of the longest communication path is not directly determined by the overall dimension of the array, as it is in conventional torus arrays. Rather, the longest communications path is limited only by the inter-cluster spacing. In one implementation, transpose elements of an N.times.N torus are combined in clusters and communicate with one another through intra-cluster communications paths. Since transpose elements have direct connections to one another, transpose operation latency is eliminated in this approach. Additionally, each PE may have a single transmit port and a single receive port. As a result, the individual PEs are decoupled from the topology of the array.

Detailed Description Text (2):
In one embodiment, a new array processor in accordance with the present invention combines PEs in clusters, or groups, such that the elements of one cluster communicate with members of only two other clusters and each cluster's constituent processing elements communicate in only two mutually exclusive directions with the processing elements of each of the other clusters. By clustering PEs in this manner, communications paths between PE clusters may be shared, thus substantially reducing the interconnection wiring required for the array. Additionally, each PE may have a single transmit port and a single receive port or, in the case of a bidirectional sequential or time sliced transmit/receive communication implementation, a single transmit/receive port. As a result, the individual PEs are decoupled from the topology of the array. That is, unlike a conventional torus connected array where each PE has four bidirectional communication ports, one for communication in each direction, PEs employed by the new array architecture need only have one port. In implementations which utilize a single transmit and a single receive port, all PEs in the array may simultaneously transmit and receive. In the conventional torus, this would require four transmit and four receive ports, a total of eight ports, per PE, while in the present invention, one transmit port and one receive port, a total of two ports, per PE are required.

CLAIMS:

5. The array processor of claim 1, wherein a cluster switch comprises said multiplexers and said cluster switch is connected to mutliplex communications received from two mutually exclusive torus directions to processing elements within a cluster.

14. The array processor of claim 10, wherein a cluster switch comprises said multiplexer and said cluster switch is connected to mutliplex communications received from two mutually exclusive torus directions to processing elements within a cluster.

WEST

Generate Collection

L9: Entry 5 of 5 File: USPT Nov 14, 1995

DOCUMENT-IDENTIFIER: US 5467455 A
TITLE: Data processing system and method for performing dynamic bus termination

Detailed Description Text (8):
The device 10 has a dynamic bus termination circuit 14 connected via at least one conductor or a bi-directional bus 13 to one or more external integrated circuit data pins. An internal data bus connects the circuit 14 to a bi-direction circuit having a first tristate buffer 22 and a second tristate buffer 24. Buffers 22 and 24 are turned on, usually in a mutually exclusive manner to enable bi-directional communication (time-multiplexed two-way communication). The buffers 22 and 24 are connected to a data unit 18 which may be a memory array or a data processor CPU. In another form the internal data bus may be split into two buses, one bus for reading and one bus for writing wherein no time multiplexing is needed until the external pins are reached.

WEST

Freeform Search

Database:
- US Patents Full-Text Database
- US Pre-Grant Publication Full-Text Database
- JPO Abstracts Database
- EPO Abstracts Database
- Derwent World Patents Index
- IBM Technical Disclosure Bulletins

Term:

Display: 10 **Documents in Display Format:** KWIC **Starting with Number** 1

Generate: ○ Hit List ◉ Hit Count ○ Side by Side ○ Image

Search | Clear | Help | Logout | Interrupt

Main Menu | Show S Numbers | Edit S Numbers | Preferences | Cases

Search History

DATE: Friday, October 17, 2003 Printable Copy Create Case